NFC Payment Spy: A Privacy Attack on Contactless Payments

Maryam Mehrnezhad, Mohammed Aamir Ali, Feng Hao, and Aad van Moorsel

School of Computing Science, Newcastle University,
Newcastle upon Tyne, United Kingdom
{m.mehrnezhad, m.a.ali2, feng.hao, aadvanmoorsel}@newcastle.ac.uk

Abstract. In a contactless transaction, when more than one card is presented to the payment terminal's field, the terminal does not know which card to choose to proceed with the transaction. This situation is called *card collision*. EMV (which is the primary standard for smart card payments) specifies that the reader should not proceed when it detects a card collision and that instead it should notify the payer. In comparison, the ISO/IEC 14443 standard specifies that the reader should choose one card based on comparing the UIDs of the cards detected in the field. However, our observations show that the implementation of contactless NFC readers in practice does not follow EMV's card collision algorithm, nor does it match the card collision procedure specified in ISO.
Due to this inconsistency between the implementation and the standards, we show an attack that may compromise the user's privacy by collecting the user's payment details. We design and implement a malicious app simulating an NFC card which the user needs to install on her phone. When she aims to pay contactlessly while placing her card close to her phone, this app engages with the terminal before the card does. Although the terminal detects a card collision (the app essentially acts like a card), it proceeds with the EMV protocol. We show the app can retrieve from the terminal the transaction data, which include information about the payment such as the amount and date. The experimental results show that our app can effectively spy on contactless payment transactions, winning the race condition caused by card collisions around 66% when testing with different cards. By suggesting these attacks we raise awareness of privacy and security issues in the specifications, standardisation and implementations of contactless cards and readers.
Keywords: NFC payment, NFC phone, Contactless payment, Privacy attack, EMV, Card collision.

1 Introduction

Near Field Communication (NFC) payment is already very popular. The statistics show that as of February 2016, £1,318.3 m was spent in the UK in the month using a contactless card. This is an increase of 19.1% on the previous month and an increase of 306.8% over the year[1]. Apart from contactless cards, other types

[1] theukcardsassociation.org.uk/contactless_contactless_statistics/

of technologies for contactless payment are suggested to the users. Examples include mobiles, tablets, watches, bPay bands, and bPay Stickers (bpay.co.uk). In fact, there are more than 350 different types of NFC-enabled devices on the market now[2].

NFC technology is based on Radio Frequency Identification (RFID) technology. Security and privacy issues of RFID communication, and in particular NFC, have been studied intensively in the literature. Contactless cards are always on and a malicious reader in the proximity of such a device is able to trigger a response from the card, without the user's awareness. A number of security and privacy violations have been reported in the literature exploiting such unauthorised readings [17]. More security attacks include different types of relay attacks such as Man-in-The-Middle and Mafia attacks [18, 21, 30, 35].

On the other hand, there are many works showing how malicious apps compromise user's security/privacy by listening to different mobile sensor data via a background process. Examples include accelerometer and gyroscope [13, 15, 22–24, 28, 36], camera and microphone [31], light [32], and Geolocation [14]. Most of these attacks work by accessing sensor data through a background process activated by a mobile app, which requires installation and user permission. Users normally install many different apps without even reviewing the app permissions. Thus, even if there is a permission request from the users, they normally ignore it [14]. This behaviour leaves the doors open for the attackers to obtain access to sensors. In this paper, we also rely on such a behaviour; we develop an app using the phone's NFC functionality which the user needs to install.

Contributions. In this paper, for the first time, we show that the NFC functionality on the victim's mobile phone can be used to compromise her contactless payment activities. This happens due to a particular situation in contactless payment which is called *card collision* or *card clash*. Card collision is the situation when more than a contactless card (or NFC-enabled device) is available in the reader's field at the same time. Card collision has been explained and addressed by EMV [10] and ISO 14443 [4][3], as the two main contactless payment references for developers to implement the contactless systems. We study these standards and propose attacks based on our findings. In particular, the contributions are:

- We explain the race condition caused by card collision and study the approaches suggested by EMV and ISO on this. We perform experiments to discover the behaviour of contactless terminals when a card collision occurs. The results show that the implementation on contactless terminals match neither EMV nor ISO.
- We show that due to this inconsistency, it is possible to track the user's contactless payment activities, for instance through a malicious app. The malicious app would have a chance to intercept payment messages and data if the phone is closely located to the contactless payment card (Fig. 1). We propose an attack vector following EMV contactless specifications by

[2] nfcworld.com/nfc-phones-list/

[3] For the rest of this paper, unless noted otherwise, by ISO standard we mean ISO/IEC 14443, and by EMV standard, we mean EMV Contactless Specifications.

Fig. 1. Different card holder cases: flip wallet, back cover/stand, Opanable back cover, sticker cover, transparent cover.

requesting the Processing Options Data Object List (PDOL) from the terminal when the malicious app wins the race and connects with the terminal first.
– We develop an Android app and perform experiments to support our claim. The results show that our attack can effectively break users' privacy and discover the pattern of their contactless payment activities.

This research highlights vulnerabilities in the standards and implementations of contactless cards and readers when a collision occurs in a contactless transaction.

2 Card Collision

In this section, first we present a real-world example of card collision which is called *Card Clash* by Transport for London (TfL) [34]. Next, we explain the approaches suggested by EMV and ISO to handle card collision.

2.1 Oystercard and Bank Card Clash

Card clash is a well-known phenomenon for a metro traveller. For example in the London metro, a traveller can either use an Oystercard or a contactless bank card[4] to pay for her journey. While swiping a wallet containing Oystercard and bank cards, the reader gets confused and does not know which card to take payment from. This causes discomfort for the users as follows [33, 34]:

– The commuter might inadvertently pay for her travel with a card she did not intend to use.
– The reader might refuse to work under this situation and the gate won't open.
– The passenger could be charged two maximum fares for the same journey. This happens when the reader charges one card when she touches in and another card when she touches out.

[4] In the rest of this paper unless noted otherwise, by bank card we mean contactless payment card.

– Even if the reader selects the contactless bank card over Oystercards for both start and end of journey, the passenger might end up being charged two times since she has already paid for a weekly travelcard on the Oystercard.

The only way to find out if any card clash happened is to sign into the user online accounts and check records of payment. If the user has been charged a maximum fare on two separate cards for the same journey, she can apply for a refund provided by TfL [34]. In fact, when TfL introduced card payments as an additional payment method to paper tickets and Oystercards in September of 2014, a huge number of double payments occurred in just a few weeks. Many of those were automatically refunded within 3-5 working days. TfL has automatically handed back about £300,000 to about 50,000 customers, with refunds averaging £5.93. Although the Card Clash issue was publicised very well, surprisingly, TFL estimates that around 1,500 instances of it are occurring every day [25]. Accordingly, different solutions have been suggested to passengers to avoid card clash [16, 20, 25, 33] which include:

– To choose the card that you want to pay with and take it out from the wallet.
– To register the Oystercard online, so that you can regularly check the online account for auditing.
– To check your bank statements regularly to find out if you have been charged on the wrong card.
– In the case of a double payment, to claim the refund by applying to the TfL website.
– To use protective cases for your contactless cards that you do not aim to pay with. Actually, Metro Bank gives free card protectors out to all of its customers.
– To switch to contactless payments. TfL has fixed the problem of weekly travelcards by applying them automatically both on Oystercards and contactless bank cards. Hence, the cost would not differ that much if a passenger switches to a contactless bank card. There are reports which show that it is even cheaper if costumers move to contactless bank cards [29].
– To use a Barclaycard contactless bPay wristband (bpay.co.uk) and pay with a wave of your hand. Any UK Visa or MasterCard debit or credit card can be linked to the bPay wristband.

Among the above solutions, those which suggest to replace the Oystercard by contactless cards or bands seem more user friendly. However, not all passengers are happy with paying for a bPay and wearing it all the time. On the other hand, people normally carry multiple bank cards. Hence even in the absence of the Oystercard, other contactless cards are still subject to card clash. Therefore, we believe that a fundamental approach is needed to overcome this real-world problem.

2.2 EMV contactless Specifications

EMV is the primary protocol standard for smart card payments. The EMV standards are managed by EMVCo (emvco.com), a consortium of multinational com-

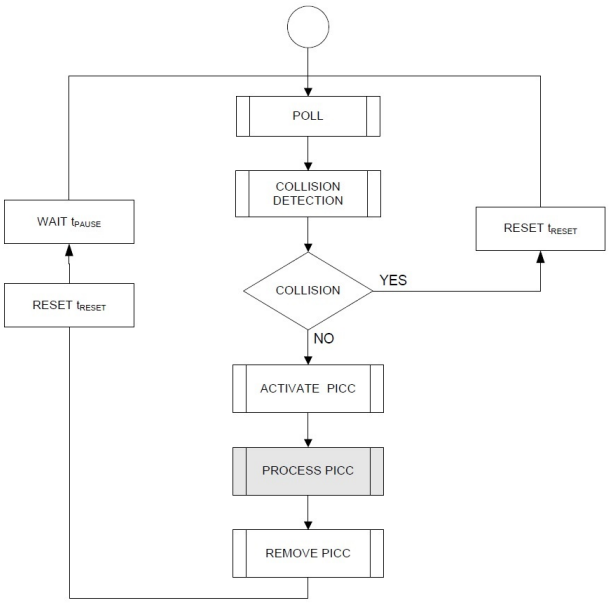

Fig. 2. Terminal Main Loop, taken from EMV contactless Book D.

panies such as Visa, MasterCard, and American Express. EMV has specifically defined specifications for contactless payment in books A, B, C and D [6–10]. ISO/IEC 14443 on the other hand, is an international standard that defines proximity cards used for identification, and the transmission protocols used for communication between the card and host. Generally, there are two ISO/IEC 14443 communication signal interfaces: Type A and Type B. They use different Radio Frequency Field (RF) modulation methods for the Proximity Coupling Device (PCD, Reader) to Proximity Integrated Circuit Card (PICC, Card) and the PICC to PCD communication. In this paper, the focus is Type A, which is the mainstream technology [26]. Android supports it, and all of our tested bank cards are Type A.

EMV Contactless Book D [10] defines *Collision* as follows: "Transmission by two or more PICCs in the same PCD energizing field and during the same time period, such that the PCD [reader] is unable to distinguish from which PICC [card] the data originated". Based on this definition, in different parts of EMV documents, the aim is to describe how EMV anti-collision mechanism handles the situation when there is more than a card in the field. Here we generally review the whole process for a contactless transaction from the reader's point of view.

According to EMV contactless Book D [10], the terminal is constantly running a main loop as illustrated in Fig 2. In the polling phase, the reader ensures that there only exists one type of technology (Type A or B) in the field by using Wake UP command e.g., WUPA for type A. Then it checks if there is only one

card from the same technology in the field. If so, it activates the card. Remember that contactless bank cards are passive, and the reader creates an energising RF (the operating field) that enables the card to power up. Next, the terminal application performs the transaction.

On the other hand, if there exists more than a card in the field, a collision is detected. Accordingly, the terminal will not initiate a transaction in this situation. The collision detection procedure is applicable either on different technologies (Type A, B, and others), or on multiple cards from the same technology. If more than one technology is in the field, the reader must report a collision, reset the operating field, and return to the polling phase. For Type A collision detection, the terminal performs a specific procedure as follows (illustrated in Fig. 5; see Appendix B). Type A cards respond to Wake UP command synchronously using Manchester coding. This allows the terminal to detect the collision in the bit level. After the terminal waits for an interval t_p, it sends WUPA command. In all parts of this algorithm, if the terminal detects a transmission error in response to the WUPA and Anti-Collision (AC) commands, it reports a collision, resets, and returns to the polling procedure. Otherwise, the reader sends an AC command which is used to obtain the complete UID of a Type A card, and to detect whether more than one Type A card is in the field. Depending on the UID size of the card, the response to the AC is different. In summary, regardless of the card collision procedure, according to EMV, **once a collision is detected, the terminal should not proceed any more; instead it should reset the field and go back to the polling procedure**.

2.3 ISO/IEC 14443

Payment cards (contact and contactless) are based on ISO/IEC 7816 [12] and ISO/IEC 14443 [2–5]. Mobile NFC payment technologies, such as Android Host-based Card Emulation (HCE) [1], are also based on ISO/IEC 14443, which is an international standard in four parts, defining the technology-specific requirements for proximity cards [2–5]. The third part of this standard [4], namely, Part 3: Initialization and anticollision, presents the same definition for collision as EMV. However, handling collision is different as we explain (presented in Fig. 6, see Appendix B).

In this standard, anticollisions are detected based on conflict in the bits of the UIDs (started from uid0 as the most significant byte). The least significant bit (LSB) of each byte is transmitted first. As an example, consider two cards as follows. Card 1: UID size = 4 bytes (single), value of uid0 =‘10’, and Card 2: UID size = 7 bytes (double). After both cards respond to the reader’s command, the terminal performs the first cascade level for the anticollision loop. As response the first card sends back the four UID bytes (uid0 uid1 uid2 uid3) plus some extra data. However since the second card’s UID is double, it sends back the cascade tag (CT) and the first three bytes (‘88’ uid0 uid1 uid2), plus some extra data. Hence the bits received in the terminal side are: (00001000)b and (00010001)b, respectively. If the implementation pads (1)b (which is what a

Card	Technology	UID size	UID0 Hex	UID0 Binary (LSB)	ISO winner
TSB visa debit- Card 1	A	4	0x35	(10101100)b	✓
TSB visa debit- Card 2	A	4	0x65	(10100110)b	✗
Barclays visa debit- Card 1	A	4	0xE7	(11100111)b	✓
Barclays visa debit- Card 2	A	4	0x87	(11100001)b	✗
barclaycard Platinum visa - Card 1	A	4	0x67	(11100110)b	✗
barclaycard Platinum visa- Card 2	A	4	0xDF	(11111011)b	✓
Nexus 5	A	4	x08	(00010000)b	✗

Table 1. Cards' information, LSB: Least Significant Bit

typical implementation does [4]) to the previous similar bits, the terminal chooses the second card over the first one and continues with it.

As it can be seen unlike EMV, **ISO specifies no termination in the case of a collision. Instead, a race condition is created in which depending on the implementation of the terminal, and the UIDs of the cards available in the field one card would be selected.** This inconsistency between EMV and ISO perhaps would cause confusion when it comes to practical implementations of these systems. We believe this is an important issue and should be addressed by the community.

3 Experiments on Contacless Readers in Practice

In this section, we examine the anticollision procedure on the contactless terminals implemented in practice. We already know that in the case of a card clash in the London metro system, the card reader may either not proceed or pick one card over others without any particular pattern [34]. It is also reported that the cards that are picked up at the start and the end of a journey may be different (in this case the passenger can apply for a refund). This explanation by TfL suggests that the implementation in practice is not consistent with either EMV or ISO. To investigate this issue further, we performed an experiment to observe how payment terminals handled card collision in practice.

Before running the experiment, we tested the NFC chipsets on the cards and the phones that we used in our experiments by writing a reader app using the getId() function[5]. Our Nexus 5 mobiles returned random 4-byte UIDs which always start with '08'. The first byte represents the brand of the technology [27]. All our tested bank cards including TSB visa debit, Barclays visa debit, and barclaycard visa have fixed 4-byte UIDs, as presented in Table 1.

In this experiment, we examined three pairs of contactless cards as presented in Table 1. Each pair has been requested and issued from the same banks and roughly at the same time. (The two TSB visa debit, and the two Barclays visa debit, were requested at the exact same time, and the two barclaycard Platinum

[5] developer.android.com/reference/android/nfc/Tag.html#getId

No.	POS	Issuing bank	Facing card to reader	Result	Note
1	MS 1, POS 1	TSB	Card 1	No operation	
2	MS 1, POS 1	TSB	Card 2	No operation	
3	MS 2, POS 1	TSB	Card 1	No operation	
4	MS 2, POS 1	TSB	Card 2	No operation	
5	MS 1, POS 2	TSB	Card 1	No operation	
6	MS 1, POS 2	TSB	Card 2	Card 1 won	msg1
7	MS 1, POS 2	TSB	Card 1	Card 2 won on 2nd try	msg1
8	MS 2, POS 2	TSB	Card 2	Card 1 won	
9	MS 2, POS 2	TSB	Card 1	No operation	
10	MS 2, POS 2	TSB	Card 1	No operation	
11	MS 1, POS 2	Barclays	Card 2	Card 1 won	
12	MS 1, POS 2	Barclays	Card 1	Card 2 won	
13	MS 1, POS 2	Barclays	Card 2	Card 1 won	msg1
14	MS 1, POS 2	Barclays	Card 1	Card 2 won	
15	MS 2, POS 1	Barclays	Card 2	Card 1 won	
16	MS 2, POS 1	Barclays	Card 1	Card 2 won	msg1
17	MS 2, POS 1	Barclays	Card 2	Card 1 won	msg1
18	MS 1, POS 3	barclaycard	Card 2	Card 1 won	
19	MS 1, POS 3	barclaycard	Card 1	Card 1 won	
20	MS 1, POS 3	barclaycard	Card 2	Card 1 won	
21	MS 1, POS 3	barclaycard	Card 1	Card 1 won	
22	MS 2, POS 2	barclaycard	Card 2	Card 1 won	
23	MS 2, POS 2	barclaycard	Card 1	Card 1 won	
24	MS 1, POS 1	barclaycard	Card 2	Card 1 won on 2nd try	msg2
25	MS 1, POS 1	barclaycard	Card 1	Card 1 won	
26	MS 2, POS 3	barclaycard	Card 2	Card 1 won	
27	MS 2, POS 3	barclaycard	Card 1	Card 1 won	

Table 2. The results of putting card pairs in the race condition. MS stands for Metro Station. In the case of No operation, the cards were presented 3 times to the POS for the same transaction. msg1: "Only present one card", msg2: "Card read failed".

visa, were requested and received within a month. The TSB card 1 had been in use more than card 2, and the barclaycard card 2 had been in use much more than card 1.) We presented each pair to different contactless terminals several times in order to put them in race conditions. We made sure that both cards were attached to each other from the same side -contactless chipsets on each other. More specifically, when tapping the cards together to the reader, we put one of the cards on top of the other one for half of the experiments, and exchanged them for the rest of the tests.

The results are presented in Table 2. As it can be seen, these results do not match the anticollision algorithms suggested by either EMV or ISO. Generally, we can not find any specific pattern in the behaviour of these terminals when facing more than a card. Interestingly, in a few cases, the terminal shows this message: "only present one card", yet it proceeds with the payment. Based

on this observation, in the next sections, we demonstrate an attack which can compromise user's privacy.

4 Attack Design

In this section, first we present the context of the attack. Then, we explain the feasibility of our attack by designing it based on the existing contactless payment specifications.

4.1 Threat Model and Attack Scenario

The context of this attack is when a user aims to pay for something by her contactless card where her card and phone are close to each other and both are presented to the reader's field. If the phone manages to hijack a few initial NFC signals that the card is meant to receive from the terminal, the attack is successful. In this situation, the phone is able to learn a lot about this contactless payment by requesting PDOL data (details in section 4.2). The data can then be sent to a remote server controlled by the attacker. However, the malicious app would not continue further communication with the reader at some point (since it does not simulate the entire payment) and the user would realise that the payment is not being processed. In order to not disappoint the user on her second effort to pay, the NFC service on the mobile should be turned off for a few minutes once it hears from the reader. In this way, the user is able to complete the payment on the second try.

There are different ways in which the user might keep her card very close to her phone. For example, different models of card holder mobile cases are available in the market now. These cases are capable of containing a few cards as shown in Fig 1. These types of wallets are already very popular with users since they offer an easy way to travel light and keep wallet essentials close to hand. When it comes to contactless payment, these accessories are even more popular since the users do not even need to take the card out of the case. Users can slide their contactless card that is kept inside the mobile case and easily tap it against the reader for daily purchases. After the increase of the cap limit from £20 to £30 in 2015, more retailer shops started to accept contactless payments for small item purchases[6].

Third parties are very interested in this sort of information, e.g., for advertising purposes. The collected information could be used in several ways. Third parties normally stimulate the users to purchase items by providing them customized ads based on this information. In addition, they can perform data mining programs to extract the patterns of people's shopping behaviours. An advanced attack might even pretend to be the user's bank by presenting this shopping information to her and tricking her to reveal her credentials via social engineering techniques. This attack in this paper can be even more impactful if

[6] theukcardsassociation.org.uk/Contactless_(our_views)/index.asp

the malicious app turns into the reader mode and extracts the card's information as suggested by Emms et al. [17]. Once the information is extracted, the app goes to the card mode for the rest of the attack. In this way, the attacker can easily pretend to be the user's bank by having her card information and her shopping records. We believe that these sorts of information are private to the users and should not be collected and shared without their permission.

4.2 Designing the Attack based on NFC Payment Protocols

In this section, we cover a few key points in relation to contactless payment protocols in which we are going to refer in our implementation. EMV Contactless Book B [7] covers the Entry Point Specification. This specification defines the reader requirements necessary to enable the discovery and selection of a contactless application, and activation of the appropriate kernel for processing the transaction. Different kernels are used for different Application Definition File (ADF) names. (e.g., for a MasterCard ADF name, Kernel 2 will be used, and for a Visa ADF name, Kernel 3 will be used.) Based on the chosen Kernel, different procedures will run to complete a payment. However, the entry point protocols are the same for all card schemes.

Entry Point is designed around the use of a Proximity Payment System Environment (PPSE) as the selection mechanism. For multi-brand acceptance, this allows a reader to quickly obtain all the available brands and applications with a single command and to make an immediate choice based on priority and kernel availability. The Entry Point command and response Application Protocol Data Units (APDUs) are presented in Fig. 3. The File Control Information (FCI) as the response to the PPSE command from the card side includes the Directory Definition File (DDF) covering a product supported by the card, the Kernel Identifier of the kernel required for the specific application underpinning the product (conditional), and the priority of the Combination (conditional). The product is indicated by its ADF name in the card. Hence, it is the card which decides what kernel to choose and talk to. Entry Point finds Combinations by matching pairs of data elements (ADF Name) and Kernel Identifier in the card with pairs of data elements in the reader (AID and Kernel ID). Once all supported Combinations have been found and the highest priority Combination has been identified, Entry Point selects the associated card application by sending a SELECT (AID) command with the ADF Name of the selected Combination. Depending on the selected AID and the kernel in the selected Combinations, a specific kernel is called to take care of the rest of the payment.

As a part of the response to SELECT AID command, the card requests Processing Options Data Object List (PDOL). Generally, a Get Processing Option (GPO) command is returned in response to this FCI command (SELECT AID) which includes the Terminal Transaction Qualifiers (TTQ), Unpredictable Number, Amount, Authorised, Transaction Currency Code, and other tags [11].

As shown Fig. 3 and we explain in the next section, our attack app is going to take the proper action in response to each command from the terminal in order to retrieve as much information as possible about each transaction.

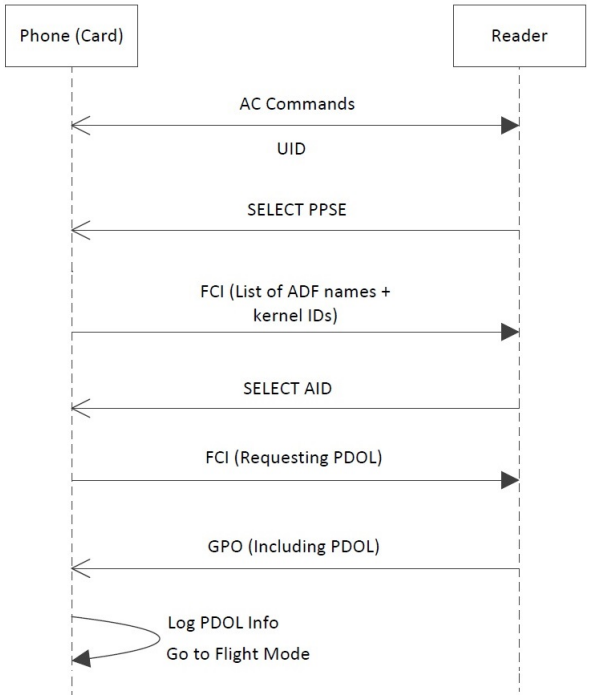

Fig. 3. The sequence diagram of the communication between our app and the reader.

5 Implementation

In this section we present the technical implementation of our attack.

5.1 Android HCE

Android supports emulating cards that are based on the NFC-Forum ISO-DEP specification (based on ISO/IEC 14443-1 to 4) and processes Application Protocol Data Units (APDUs) as defined in the ISO/IEC 7816-4 specification. In compliance with ISO/IEC 7816-4, each HCE application has an Application ID (AID). This ID enables the reader app to select the correct service.

In our implementation, we declared an AID group including an AID filter of a Visa card (0xA0000000031010) in an XML resource to be pointed by a `SERVICE_META_DATA` tag in the manifest declaration. On the other hand, Android does not interpret the PPSE selection command and, consequently, it does not generate or send a list of available payment applications. Hence we have to handle the PPSE command in the app. Typically, an HCE payment application based on EMV standards would register for both: the payment application AID and the PPSE ADF name. Note that from a protocol perspective there is no difference between an ADF name and an AID, so we can register for it in our

service XML file with an AID filter for the ADF name ("2PAY.SYS.DDF01") in its ASCII hexadecimal representation of 0x325041592E5359532E4444463031. In the same file, we set the `android:requireDeviceUnlock` attribute to `false` in order to avoid the user being asked for unlocking her device.

The `HostApduService` class is extended for implementing an HCE service with two abstract methods: `processCommandApdu` and `onDeactivated`. The former is called whenever the card receives an APDU from an NFC reader and enables half-duplex communication with the reader. The latter is called when either the NFC link is broken or the reader wishes to talk to another service. According to EMV, the first two APDUs (SELECT PPSE and SELECT AID) are for service selection. That is where we request PDOL, as shown in Fig. 3. After a successful service selection, the card and reader can exchange any type of data. When the app receives the first GPO command including the requested data, it logs the data in a file, and the attack terminates. Accordingly, our app turns the NFC off by going to the flight mode to allow the user to complete the purchase on the second try.

5.2 Android Flight Mode

Android does not offer any API for turning the NFC controller on/off programmatically. Therefore, developers usually set the NFC settings in a way that prompts the user to turn it on/off manually. In our attack, once our app hears from the terminal, it needs to turn off the NFC, so that the user can successfully pay on her second try. One possible way to control the NFC adapter is to change the phone's airplane mode setting. However, only those apps with the superuser permissions can change the Airplane mode setting which requires `WRITE_SETTINGS` and `WRITE_SECURE_SETTINGS` to be declared in the manifest file. Starting from Android 4.2 , turning on/off airplane mode is not supported by android APIs any more. Hence, this part of our attack only works on a rooted device.

On the other hand, this attack needs to keep the phone's screen on since, at the moment, NFC does not work when the phone is off [1]. An advanced attack would turn the screen on only when the user wants to pay by using accelerometer and gyroscope sensor measurements in order to recognise such a gesture. Li et. al. in [19] show that it is effectively possible to use the tap gesture to unlock the phone for NFC applications based on accelerometer data. By augmenting such a gesture recognition feature to our code, we will have a complete application that is able to compromise users privacy in contactless payments.

6 Experiments and Results

We performed an experiment by installing the app on Android phones (Nexus 5). We attached the card to the back of the phone in two different positions, as shown in Fig 4. The position that the card was attached to the phone was important in our experiments since it effected the results, as we explain later.

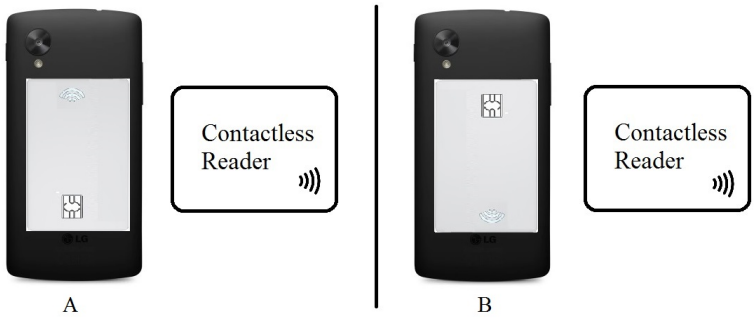

Fig. 4. Contactless card attached to the phone in two different positions for the experiments; A (left): the NFC chipset was down, B (right): the NFC chipset was up.

In all experiments, the back of the phone was faced to the terminal (hence, the card was in a closer distance to the terminal than the phone).

6.1 Expected Results

According to the EMV specifications, regardless of the UID of the card and the phone, the terminal should not proceed in the case of a card collision. ISO standards, however, suggest to opt one of the UIDs (typically with higher values) in the race condition. The first UID byte (UID0) of mobile phones that we tested is always '08' (LSB: 00010000), and it is a single UID. As presented in Table 1, all of our cards should always win over the phone if it is a typical ISO implementation. In the following experiments, we show that the expected behaviour does not happen in practice, and the phone wins with a high probability.

6.2 Experiment A: Card and Phone Collision

In this experiment, we tested a few different contactless cards by presenting each card with the phone to a few terminals (contactless metro ticket machines). We tested multiple cards including two TSB visa debit, and two Barclays visa debit on different machines. During this experiment, we asked a few users to pay for metro tickets with different contactless cards that we provided to them. These cards were attached to mobile phones (Nexus 5). These participants were informed of the purpose of the experiment, but were not asked to follow any particular procedure. We asked them to naturally pay contactlessly. We observed the behaviour of the terminals as summarized in Table 4 (see Appendix A).

The results show that when the card is attached to the phone in position A (the card's NFC chipset is down), the phone can hear the reader's signal first with a very high probability. On the other hand, when the card's NFC chipset is positioned to the top of the phone (position B), the chance of the card winning is slightly more than the phone. Nevertheless, an average user might put the card in any of these two positions close to the phone. Based on our experiment,

Sender	APDU	Command
Terminal	00A404000E325041592E5359532E444446303100	SELECT PPSE
Phone	6F3C840E325041592E5359532E4444463031A52A BF0C2761254F07A0000000031010870101501042 4152434C4159434152442056495341BF6304DF20 01809000	FCI
Reader	00A4040007A000000003101000	SELECT AID
Phone	6F4B8407A0000000031010A5405010424152434C 4159434152442056495341870101**9F38**189F6604 **9F02**069F03069F1A0295055F2A02**9A**039C019F37 045F2D02656EBF0C089F5A0531082608269000	FCI including PDOL request
Terminal	80A8000023**83**2130000000**000000000080**000000 0000000826000000000000826**160523**0016126739 00	GPO including PDOL data

Table 3. Exchanged APDUs of experiment B

generally our app is able to recognise about 66% of the user's contactless payment activities. Over time, this success rate would allow the attacker to accumulate information about the user's contactless payment patterns.

Our observations show that contactless terminals present different messages on the display based on the situation. When select to pay, it displays: "Insert, swipe or tap for GBP 0.80" as the first message. If it can not choose either the card or the phone it displays: "Card read failed", and it goes back to the first message. The fail message happened when our users tapped the card and the phone very quickly, hence none of them were presented to the field for a sufficient time. Similar to our experiments in Section 3, the terminal may show another message: "Only present one card", but it still proceeds with the transaction.

6.3 Experiment B: PDOL Data

In order to show the impact of the attack more visibly, we performed another experiment. While purchasing a ticket, we presented our final app to a payment terminal in a metro station. Our app logged the PDOL data of the transaction and then went to the Airplane mode. We built our card app in a way that it responded to the two SELECT commands – PPSE and AID – before asking for PDOL data (see Fig. 3).

The exchanged command and response APDUs are shown in Table 3. As it can be seen, when the card sends the second FCI, by sending PDOL tag ('9F38'), it requests different sort of information about the transaction such as the amount (tag='9F02', Amount, Authorised (Numeric)) and transaction date (tag='9A'). Accordingly, the terminal responds with the first GPO command including the requested items for PDOL ('83') i.e. amount ('000000000080' = 0.80 pence) and date ('160523' = 2016 May 23) [11].

As it can be seen, the attacker can easily build such a table for all transactions and discover the user's payment patterns.

7 Conclusion

In this paper, we discussed a real world problem concerning the card collision when making contactless payments. We studied the EMV and ISO standards on card collision, and by performing experiments we discovered that the implementation in practice matches neither of them. Based on this inconsistency, we describe and implement an attack on the privacy of contactless payments. In this attack, we simulated a card within an app and tracked the user's contactless payment transactions by requesting PDOL data from the terminal. When the phone and the card are both presented to a contactless terminal, our app could successfully win the race condition over the card in the majority of test cases.

Our findings suggest vulnerabilities in the current infrastructure which needs to be addressed. More specifically, the results of our experiments show that when tapping the terminal with more than a card, in most cases (Tables 2 and 4), the terminal does not even identify the card collision. Nevertheless, even if the terminal identifies the presence of more than a card in the field (by showing a message), it still proceeds with the transactions. The selection of the card appears random. A countermeasure to this identified privacy attack is updating the implementation of the payment terminals according to EMV's card collision algorithm: i.e., the process should not proceed when more that one card is detected in the field. Updating some parts of EMV's protocol and protecting the PDOL data would also mitigate the introduced attack. Finally, the EMV and ISO standards would need to be updated to have a consistent algorithm to handle card collision.

Acknowledgement

We would like to thank Dr. Michael Ward from EMV and Digital Devices for his valuable help towards our better understanding of EMV contactless specifications. We would like to thank Dr. Martin Emms and Mr. Ehsan Toreini from Newcastle University for their help on performing the experiments of this work. We also thank all the anonymous reviewers of this paper. All experiments gained approval through Newcastle University's research ethics processes. Feng Hao was supported by ERC Starting Grant No 306994, Aad van Moorsel was supported by EPSRC grant K006568.

References

1. Host-based card emulation. Available online at http://developer.android.com/guide/topics/connectivity /nfc/hce.html.
2. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, BS ISO/IEC 14443-1:2008+A1:2012 Identification cards. Contactless integrated circuit cards. Proximity cards. Physical characteristics, 2012. Available at www.bsol.bsigroup.com.

3. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, BS ISO/IEC 14443-2:2010+A2:2012 Identification cards. Contactless integrated circuit cards. Proximity cards. Radio frequency power and signal interface, 2012. Available at www.bsol.bsigroup.com.
4. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, BS ISO/IEC 14443-3:2011+A6:2014 Identification cards. Contactless integrated circuit cards. Proximity cards. Initialization and anticollision, 2014. Available at www.bsol.bsigroup.com.
5. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, BS ISO/IEC 14443-4:2008+A4:2014 Identification cards. Contactless integrated circuit cards. Proximity cards. Transmission protocol , 2014. Available at www.bsol.bsigroup.com.
6. EMV Contactless Specifications for Payment Systems, Book A: Architecture and General Requirements, 2015. Available at www.emvco.com /specifications.aspx?id=21.
7. EMV Contactless Specifications for Payment Systems, Book B: Entry Point, 2015. Available at www.emvco.com/specifications.aspx?id=21.
8. EMV Contactless Specifications for Payment Systems, Book C2: Kernel 2 Specification, 2015. Available at www.emvco.com/specifications.aspx?id=21.
9. EMV Contactless Specifications for Payment Systems, Book C3: Kernel 3 Specification, 2015. Available at www.emvco.com/specifications.aspx?id=21.
10. EMV Contactless Specifications for Payment Systems, Book D: Contactless Communication Protocol, 2015. Available at www.emvco.com /specifications.aspx?id=21.
11. EMV Integrated Circuit Card Specifications for Payment Systems, Book 3, 2011. Available at www.emvco.com/specifications.aspx?id=223.
12. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, BS ISO/IEC 7816-4:2013, Identification cards. Integrated circuit cards. Organization, security and commands for interchange, 2013. Available atwww.bsol.bsigroup.com.
13. A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith. Practicality of accelerometer side channels on smartphones. In *Proceedings of the 28th Annual Computer Security Applications Conference*, pages 41–50. ACM, 2012.
14. R. Balebako, J. Jung, W. Lu, L. F. Cranor, and C. Nguyen. "little brothers watching you": Raising awareness of data leaks on smartphones. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS '13, pages 12:1–12:11, New York, NY, USA, 2013. ACM.
15. L. Cai and H. Chen. Touchlogger: Inferring keystrokes on touch screen from smartphone motion. In *HotSec*, 2011.
16. M. Curphey. Card clash, what is it, and how to avoid ir, 2014. Available online at http://uk.creditcards.com/credit-card-news/what-is-card-clash-and-how-to-avoid-it-1372.php.
17. M. Emms, B. Arief, N. Little, and A. van Moorsel. Risks of offline verify pin on contactless cards. In *Financial Cryptography and Data Security*, volume 7859 of *Lecture Notes in Computer Science*, pages 313–321. Springer Berlin Heidelberg, 2013.
18. T. Halevi, D. Ma, N. Saxena, and T. Xiang. *Computer Security – ESORICS 2012: 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10-12, 2012. Proceedings*, chapter Secure Proximity Detection for NFC Devices Based on Ambient Sensor Data, pages 379–396. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

19. H. Li, D. Ma, N. Saxena, B. Shrestha, and Y. Zhu. Tap-wave-rub: Lightweight malware prevention for smartphones using intuitive human gestures. In *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '13, pages 25–30, New York, NY, USA, 2013. ACM.
20. G. Marshall. Travel using contactless cards: An update from tfl, 2014. Available online at http://londonist.com/2014/07/travel-using-contactless-cards-an-update-from-tfl.
21. M. Mehrnezhad, F. Hao, and S. F. Shahandashti. *Security Standardisation Research: Second International Conference, SSR 2015, Tokyo, Japan, December 15-16, 2015, Proceedings*, chapter Tap-Tap and Pay (TTP): Preventing the Mafia Attack in NFC Payment, pages 21–39. Springer International Publishing, Cham, 2015.
22. M. Mehrnezhad, E. Toreini, S. F. Shahandashti, and F. Hao. Touchsignatures: Identification of user touch actions based on mobile sensors via javascript. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ASIA CCS '15, pages 673–673, New York, NY, USA, 2015. ACM.
23. M. Mehrnezhad, E. Toreini, S. F. Shahandashti, and F. Hao. Touchsignatures: Identification of user touch actions and pins based on mobile sensor data via javascript. *Journal of Information Security and Applications*, 26:23 – 38, 2016.
24. E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury. Tapprints: your finger taps have fingerprints. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 323–336. ACM, 2012.
25. K. Morley. Contactless cards: how to avoid paying twice, 2014. Available online at http://www.telegraph.co.uk/finance/personalfinance/money-saving-tips/11215583/Contactless-cards-how-to-avoid-paying-twice.html.
26. ISO 14443, ISO 18092, Type-A, Type-B, Type-F, Felica, Calypso NFCIP, NFC-HELP!, 2009. Available online at http://www.nfc.cc/2009/01/03/iso-14443-iso-18092-type-a-type-b-type-f-felica-calypso-nfcip-nfc-help/.
27. AN10927, MIFARE and handling of UIDs, 2013. By NXP, Company Public.
28. E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang. Accessory: password inference using accelerometers on smartphones. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, page 9. ACM, 2012.
29. H. Saul. Oyster card users pay up to £91 more each week than people using new contactless payment, 2014. Available online at http://www.independent.co.uk/news/uk/home-news/oyster-card-users-pay-up-to-91-more-each-week-than-people-using-new-contactless-payment-9843642.htmll.
30. B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan. *Drone to the Rescue: Relay-Resilient Authentication using Ambient Multi-sensing*, pages 349–364. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
31. L. Simon and R. Anderson. Pin skimmer: Inferring pins through the camera and microphone. In *Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '13, pages 67–78, New York, NY, USA, 2013. ACM.
32. R. Spreitzer. Pin skimming: Exploiting the ambient-light sensor in mobile devices. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '14, pages 51–62, New York, NY, USA, 2014. ACM.
33. Why contactless cards can leave you with a losing deal, 2014. Available online at http://www.theguardian.com/money/2013/may/25/contactless-cards.
34. Watch out for card clash. Available online at https://tfl.gov.uk/fares-and-payments/contactless/card-clash.

35. J. Vila and R. J. Rodríguez. *Radio Frequency Identification: 11th International Workshop, RFIDsec 2015, New York, NY, USA, June 23-24, 2015, Revised Selected Papers*, chapter Practical Experiences on NFC Relay Attacks with Android, pages 87–103. Springer International Publishing, Cham, 2015.
36. Z. Xu, K. Bai, and S. Zhu. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, pages 113–124. ACM, 2012.

A Experiment Results

In this section, we provide the detailed results of our Card and Phone Collision experiment. These results are presented in Table 4.

B EMV and ISO Flowcharts

The collision detection procedure of EMV specification and Anticollision loop flowchart of ISO are presented in Fig. 5 and Fig. 6, respectively.

No.	Card	Terminal	Position	Winner	Msg
1	TSB 1	MS 1, POS 2	A	Phone	
2	TSB 1	MS 2, POS 2	A	Phone	
3	TSB 1	MS 2, POS 2	A	Card	
4	TSB 1	MS 2, POS 2	A	Phone	
5	TSB 1	MS 1, POS 1	B	Card	
6	TSB 1	MS 1, POS 1	B	Card	
7	TSB 1	MS 1, POS 1	B	Phone	
8	TSB 1	MS 1, POS 1	B	Phone	
9	TSB 1	MS 2, POS 2	B	Card	
10	TSB 1	MS 2, POS 2	B	Card	
11	TSB 2	MS 1, POS 2	A	Phone	
12	TSB 2	MS 1, POS 2	A	Phone	
13	TSB 2	MS 1, POS 2	A	Phone	
14	TSB 2	MS 1, POS 2	A	Phone	
15	TSB 2	MS 1, POS 2	A	Phone	
16	TSB 2	MS 3, POS 1	A	Phone	
17	TSB 2	MS 3, POS 2	B	Card	
18	TSB 2	MS 3, POS 2	B	Phone	
19	TSB 2	MS 3, POS 2	B	Phone, 2nd try	“Card read failed”
20	TSB 2	MS 3, POS 2	B	Card, 2nd try	“Card read failed”
21	TSB 2	MS 3, POS 2	B	Phone	
22	Barclays 1	MS 1, POS 1	A	Phone	
23	Barclays 1	MS 1, POS 1	A	Phone	
24	Barclays 1	MS 1, POS 1	A	Phone, 2nd try	“Card read failed”
25	Barclays 1	MS 1, POS 1	A	Phone	
26	Barclays 1	MS 1, POS 1	A	Phone	
27	Barclays 1	MS 1, POS 1	A	Phone	
28	Barclays 1	MS 1, POS 1	B	Card	
29	Barclays 1	MS 1, POS 1	B	Phone	
30	Barclays 1	MS 1, POS 2	B	Card, 2nd try	“Card read failed”
31	Barclays 1	MS 1, POS 2	B	Phone	
32	Barclays 1	MS 1, POS 2	B	Card	
33	Barclays 1	MS 1, POS 2	B	Phone	
34	Barclays 2	MS 1, POS 2	A	Phone	
35	Barclays 2	MS 1, POS 2	A	Phone	
36	Barclays 2	MS 1, POS 2	A	Phone	
37	Barclays 2	MS 1, POS 2	A	Phone	
38	Barclays 2	MS 1, POS 2	A	Card	“Only present one card”
39	Barclays 2	MS 1, POS 2	B	Card	“Only present one card”
40	Barclays 2	MS 1, POS 2	B	Card, 2nd try	“Card read failed”
41	Barclays 2	MS 1, POS 2	B	Phone	
42	Barclays 2	MS 1, POS 1	B	Card	
43	Barclays 2	MS 1, POS 1	B	Card	
44	Barclays 2	MS 1, POS 1	B	Phone, 2nd try	“Card read failed”

Table 4. Results of experiment A

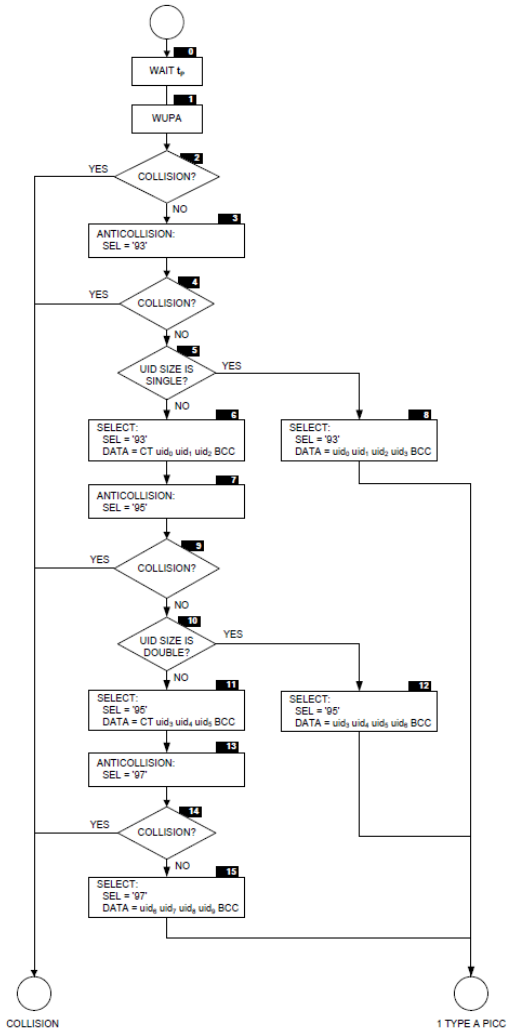

Fig. 5. Type A collision detection, taken from EMV contactless Book D.

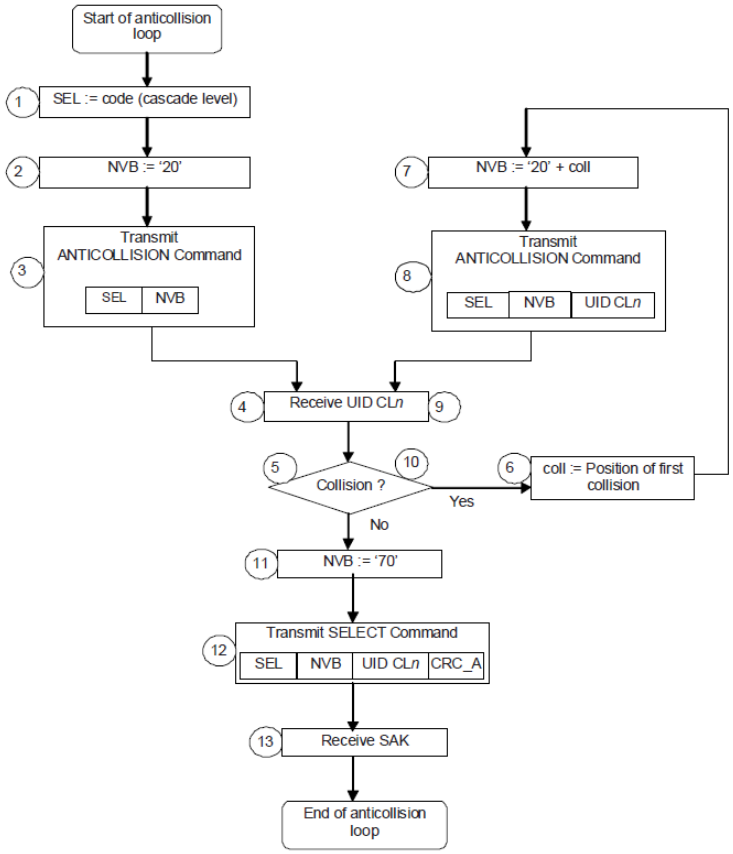

Fig. 6. Anticollision loop, flowchart for PCD, taken from ISO/IEC 14443-3.